HTTP Bad Download

Arsyad Dwiyankuntoko
11ipa3.arsyad@gmail.com
http://arsyaddwiyankuntoko.blogspot.com

Lisensi Dokumen:
Copyright © 2003-2007 IlmuKomputer.Com
Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

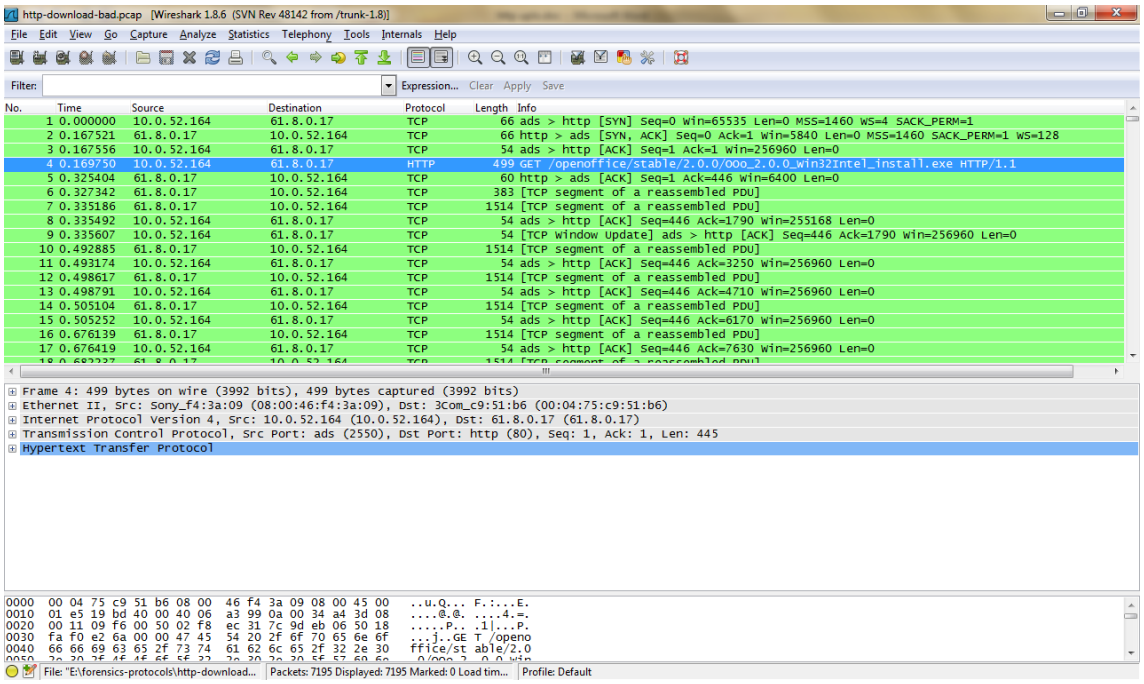
Pendahuluan

Pada jaman sekarang ini informasi sangat mudah didapat dengan adanya internet. Apapun bisa didapat mulai dari hal yang umum sampai hal yang bersifat rahasia sekalipun. Orang-orang dapat dengan mudah mencari informasi yang mereka inginkan hanya dengan mengetikan informasi yang mereka inginkan itu saja. hal tersebut tidak akan pernah bisa terjadi tanpa adanya peran dari server yang melayani user. Jadi sebenarnya pada saat user mengetikan informasi yang mereka cari, user telah melakukan request ke server dan server-lah yang memberikan informasi yang diinginkan user tadi. Untuk bisa melakukan koneksi dari user ke server pun terdapat suatu protokol tersendiri yang sudah popular yaitu HTTP. HTTP (Hypertext Transfer Protocol) merupakan suatu protokol jaringan komputer yang berfungsi untuk sistem informasi terdistribusi, kolaboratif, dan menggunakan hipermedia. Dengan adanya HTTP user dapat melakukan berbagai macam aktifitas yang sumber datanya berasal dari server seperti browsing, download, upload, streaming, dll.

Meskipun begitu, sering kali aktifitas yang dilakukan melalui internet tersebut mengalami kendala dimana salah satu kendalanya adalah download yang jelek. Pada tulisan ini akan dicoba dilakukan analisis terhadap hasil capture http-download-bad.pcap yang telah di-capture oleh Agus Kurbiawan. File tersebut bisa di-download di if-unpas.org/iwank/Wireshark/network forensic agus kurbiawan

Download merupakan suatu proses pengiriman data dari server ke user dimana data yang dikirimkan tersebut baru bisa digunakan setelah proses download selesai.

Sewaktu file di-download dari server ke user, file tersebut tidak dikirim secara utuh dalam sekali kirim tetapi file tersebut dibagi-bagi menjadi beberapa paket dan dikirim secara berurutan ke tujuan. Protokol yang bertugas dalam pengiriman paket ini adalah protokol pada layer transport HTTP, yaitu TCP. HTTP merupakan protocol aplikasi yang memungkinkan komunikasi antara client dan user. Berikut merupakan contoh hasil capture dari proses download



Pada gambar tersebut terlihat jika 10.0.52.164 merupakan client yang sedang mencoba meminta file ke server 61.8.0.17. hal tersebut terlihat pada paket 4 dengan protocol HTTP yang berisi command GET dari client ke server yang berarti client sedang meminta file pada resource tertentu. Sebelum melakukan GET, client dan server harus melakukan proses handshaking terlebih dahulu untuk menjalin sebuah koneksi. Pada gambar pada paket 1,2, dan 3 merupakan proses 3-way handshaking tersebut. Disitu terlihat jika client mengirim SYN terlebih dahulu kemudian server menjawabnya dengan mengirim SYN, ACK dan client kembali menjawabnya dengan ACK. 3-way handshaking ini dilakukan untuh menegosiasikan parameter-parameter apa saja yang akan digunakan pada saat pengiriman data yang akan dilakukan nanti. Setelah proses handshaking selesai dan client sudah melakukan request terhadap file yang diinginkan,

Copyright © 2003-2007 IlmuKomputer.Com

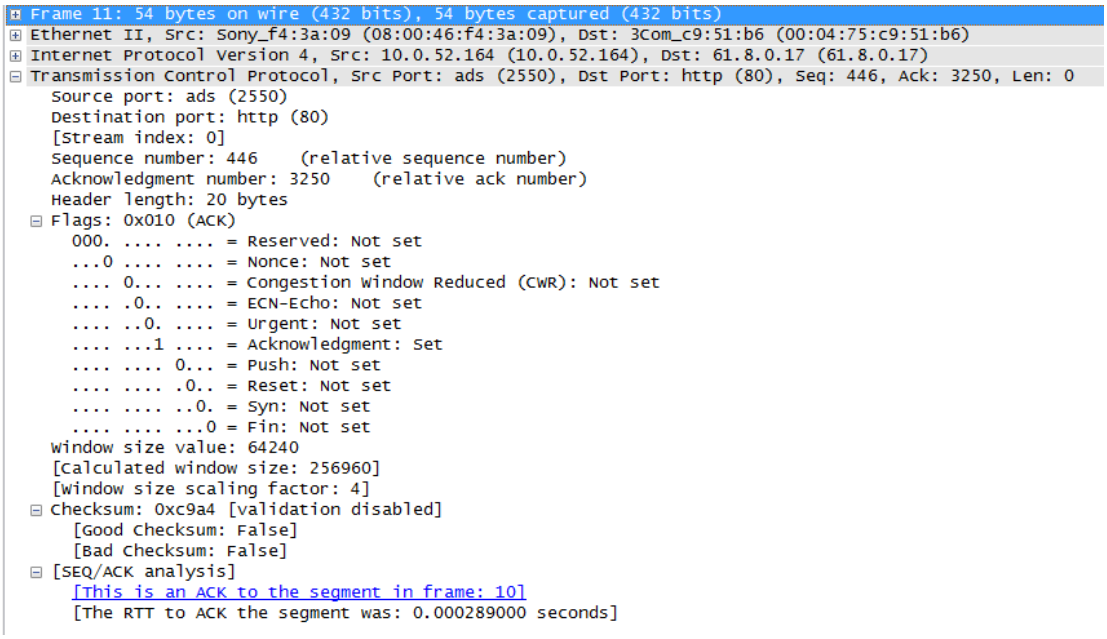
maka file tersebut akan langsung dikirim ke client. Seperti yang sudah dijelaskan tadi, file tersebut akan dibagi menjadi beberapa paket dan dikirim secara berurutan. Pada hasil capture yang didapat dapat dilihat jika server mengirim banyak paket ke client. Pada gambar terlihat bahwa paket yang dikirim ke client terdapat info yang berisi *TCP segment of reassembled PDU* yang berarti segment dari TCP dari PDU (Packet Data Unit) yang sedang dikumpulkan kembali. Selain itu setelah paket dikirim ke client, client selalu mengirim pesan ACK ke server sebagai reply-nya. Jadi server tidak akan mengirim paket selanjutnya apabila client belum mengirim ACK ke server. Proses tersebut akan terus menerus terjadi sampai paket yang dikirim membentuk suatu file yang utuh. Berikut merupakan contoh detail isi paket yang dikirim pada hasil capture tersebut

```
⊞ Frame 10: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
⊞ Ethernet II, Src: 3Com_c9:51:b6 (00:04:75:c9:51:b6), Dst: Sony_f4:3a:09 (08:00:46:f4:3a:09)
⊞ Internet Protocol Version 4, Src: 61.8.0.17 (61.8.0.17), Dst: 10.0.52.164 (10.0.52.164)
⊟ Transmission Control Protocol, Src Port: http (80), Dst Port: ads (2550), Seq: 1790, Ack: 446, Len: 1460
    Source port: http (80)
    Destination port: ads (2550)
    [Stream index: 0]
    Sequence number: 1790    (relative sequence number)
    [Next sequence number: 3250    (relative sequence number)]
    Acknowledgment number: 446    (relative ack number)
    Header length: 20 bytes
  ⊟ Flags: 0x010 (ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
    Window size value: 50
    [Calculated window size: 6400]
    [Window size scaling factor: 128]
  ⊟ Checksum: 0x3a4d [validation disabled]
      [Good Checksum: False]
      [Bad Checksum: False]
  ⊟ [SEQ/ACK analysis]
      [Bytes in flight: 1460]
    TCP segment data (1460 bytes)
```

Gambar diatas merupakan contoh detail paket yang dikirim oleh server ke client. Disitu terdapat berbagai macam parameter yang digunakan seperti source dan destination port, sequence number, flags, dll. Pada setiap paket yang dikirim dari server memiliki sequence number yang berbeda-beda dan pada paket tersebut juga terdapat keterangan next sequence number sehingga data yang dikirimkan nantinya dapat diurutkan agar membentuk suatu file. Selain itu, pada paket yang dikirim dari server tersebut juga terdapat acknowledgment number yang sudah ditetapkan. jadi, setelah paket tersebut dikirimkan akan ada reply acknowledgement dari client. Berikut detail dari paket

Copyright © 2003-2007 IlmuKomputer.Com

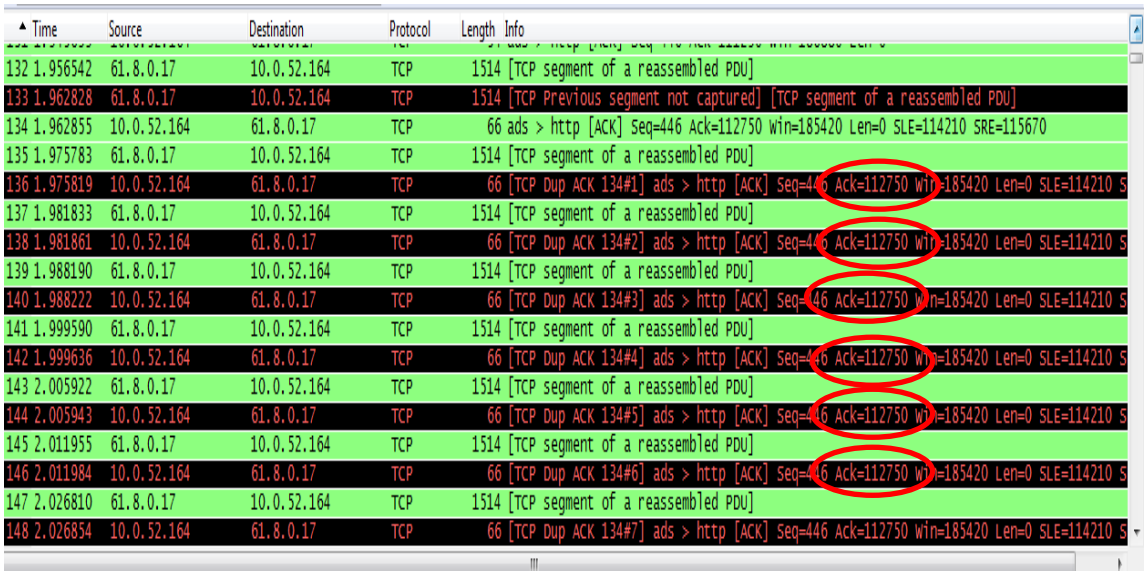
acknowledgement yang dikirim client sebagai respon dari paket yang dikirim dari server tersebut



Pada gambar tersebut terlihat jika acknowledgment number adalah 3250, sama dengan sequence number paket yang dikirim dari server sebelumnya. Hal tersebut membuktikan jika paket tersebut merupakan ACK dari paket pada frame 10 seperti yang sudah dijelaskan tadi. Pada SEQ/ACK analysis juga tertulis jika paket ACK tersebut merupakan ACK dari segment pada frame 10. Apabila paket ACK ini telah diterima oleh server, barulah server akan mengirim paket selanjutnya.

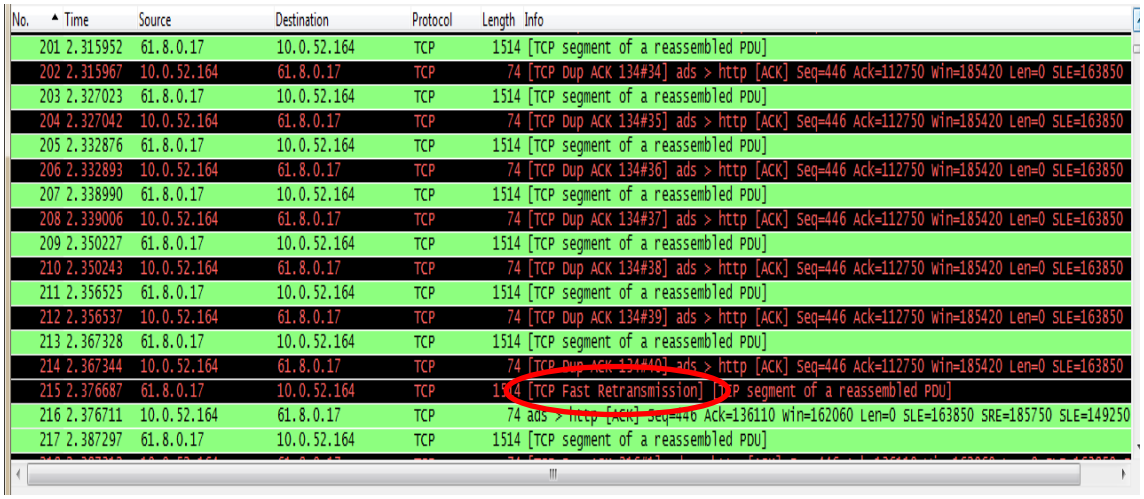
Meskipun begitu, terkadang paket yang dikirim oleh server tidak selalu bisa diterima dengan baik oleh client. Paket tersebut tidak diterima karena berbagai macam hal seperti firewall, koneksi internet yang lambat, adanya third-party, dll. Apabila ada paket yang hilang maka client tidak akan bisa menyusun kembali paket-paket yang lain menjadi sebuah file yang utuh. Oleh karena itu, saat ada paket yang hilang server harus mengirim ulang paket tersebut ke client. Agar server mengetahui jika ada paket yang hilang, client memberitahunya dengan cara terus mengirim duplicate ACK atau ACK dengan number yang sama secara terus menerus sampai server melakukan fast retransmission dari paket yang hilang tadi. Jadi, duplicate ACK ini menjadi trigger yang akan membuat server mengetahui jika ada paket yang tidak berhasil terkirim. Berikut merupakan contoh duplicate ACK yang dikirim ke server.

Copyright © 2003-2007 IlmuKomputer.Com



Time	Source	Destination	Protocol	Length	Info
132 1.956542	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
133 1.962828	61.8.0.17	10.0.52.164	TCP	1514	[TCP Previous segment not captured] [TCP segment of a reassembled PDU]
134 1.962855	10.0.52.164	61.8.0.17	TCP	66	ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=114210 SRE=115670
135 1.975783	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
136 1.975819	10.0.52.164	61.8.0.17	TCP	66	[TCP Dup ACK 134#1] ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=114210 S
137 1.981833	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
138 1.981861	10.0.52.164	61.8.0.17	TCP	66	[TCP Dup ACK 134#2] ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=114210 S
139 1.988190	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
140 1.988222	10.0.52.164	61.8.0.17	TCP	66	[TCP Dup ACK 134#3] ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=114210 S
141 1.999590	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
142 1.999636	10.0.52.164	61.8.0.17	TCP	66	[TCP Dup ACK 134#4] ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=114210 S
143 2.005922	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
144 2.005943	10.0.52.164	61.8.0.17	TCP	66	[TCP Dup ACK 134#5] ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=114210 S
145 2.011955	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
146 2.011984	10.0.52.164	61.8.0.17	TCP	66	[TCP Dup ACK 134#6] ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=114210 S
147 2.026810	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
148 2.026854	10.0.52.164	61.8.0.17	TCP	66	[TCP Dup ACK 134#7] ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=114210 S

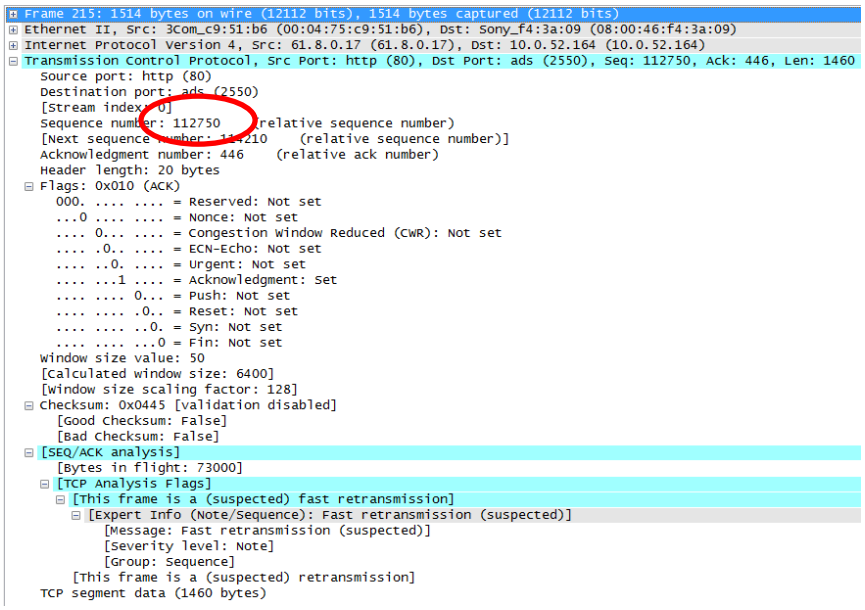
Berdasarkan gambar tersebut, terlihat jika pada paket 133 terdapat keterangan yang menyatakan jika paket yang dikirim tidak tercapture atau tidak diterima oleh client. Setelah itu client akan merespon dengan tetap mengirim ACK ke server. Pada gambar tersebut ACK dengan nomer yang sama tidak hanya dikirim sekali melainkan berkali-kali hingga server melakukan fast retransmission seperti gambar di bawah ini



No.	Time	Source	Destination	Protocol	Length	Info
201	2.315952	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
202	2.315967	10.0.52.164	61.8.0.17	TCP	74	[TCP Dup ACK 134#34] ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=163850
203	2.327023	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
204	2.327042	10.0.52.164	61.8.0.17	TCP	74	[TCP Dup ACK 134#35] ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=163850
205	2.332876	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
206	2.332893	10.0.52.164	61.8.0.17	TCP	74	[TCP Dup ACK 134#36] ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=163850
207	2.338990	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
208	2.339006	10.0.52.164	61.8.0.17	TCP	74	[TCP Dup ACK 134#37] ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=163850
209	2.350227	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
210	2.350243	10.0.52.164	61.8.0.17	TCP	74	[TCP Dup ACK 134#38] ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=163850
211	2.356525	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
212	2.356537	10.0.52.164	61.8.0.17	TCP	74	[TCP Dup ACK 134#39] ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=163850
213	2.367328	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]
214	2.367344	10.0.52.164	61.8.0.17	TCP	74	[TCP Dup ACK 134#40] ads > http [ACK] Seq=446 Ack=112750 Win=185420 Len=0 SLE=163850
215	2.376687	61.8.0.17	10.0.52.164	TCP	1514	[TCP Fast Retransmission] [TCP segment of a reassembled PDU]
216	2.376711	10.0.52.164	61.8.0.17	TCP	74	ads > http [ACK] Seq=446 Ack=136110 Win=162060 Len=0 SLE=163850 SRE=185750 SLE=149250
217	2.387297	61.8.0.17	10.0.52.164	TCP	1514	[TCP segment of a reassembled PDU]

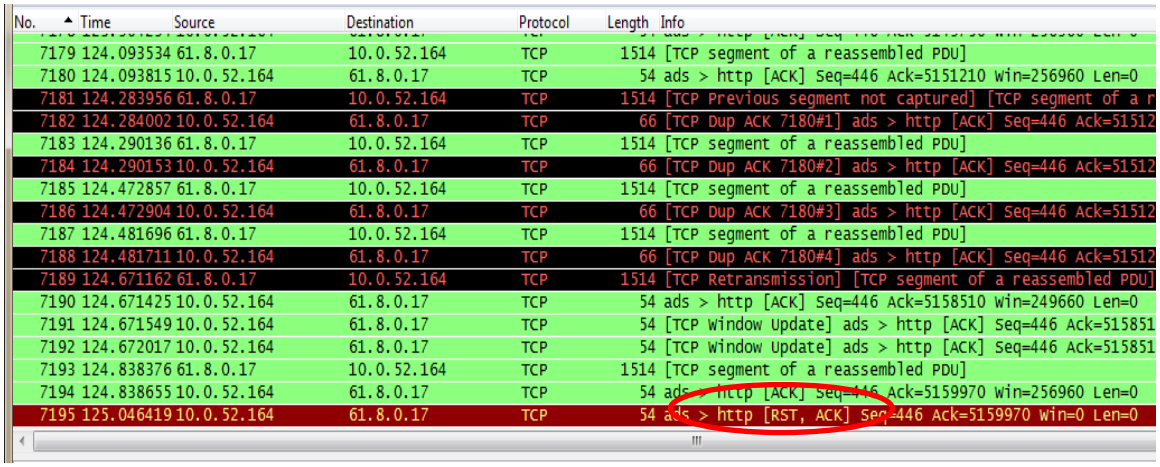
Pada gambar tersebut terlihat jika client telah mengirim sebanyak 40 duplicate ACK dan server baru mengirimkan ulang paket yang diinginkan oleh client tersebut. Pada paket 216 yang terletak dibawah paket fast retransmission terlihat jika ACK yang dikirim nomernya sudah berbeda dari duplicate ACK sebelumnya. Itu berarti client sudah menerima paket fast retransmission tadi dan menginginkan paket selanjutnya untuk dikirim. Untuk lebih jelasnya, berikut merupakan gambar dari detail paket fast retransmission.

Copyright © 2003-2007 IlmuKomputer.Com



Pada gambar tersebut terlihat jika sequence number-nya sama dengan ACK number dari duplicate ACK yang terus menerus dikirim tadi. Hal tersebut membuktikan jika server telah berhasil mengirim paket yang diinginkan oleh client.

Selain masalah paket yang tidak dapat diterima tersebut, pada akhir capture yang dilakukan terdapat sebuah paket yang berisi RST, ACK seperti gambar di bawah ini.



Berdasarkan berbagai sumber yang telah dicari, RST ini muncul bisa jadi dikarenakan oleh berbagai macam hal, yaitu:

- Server telah menutup port yang digunakan untuk melakukan koneksi dengan client saat proses download sedang berjalan
- Client kehilangan koneksi dengan server
- Adanya *malformed packet* (paket yang cacat)

Copyright © 2003-2007 IlmuKomputer.Com

Referensi

http://if-unpas.org/iwank/Wireshark/network%20forensik%20wireshark%20agus%20kurbiawan/forensics-protocols/
http://www.wireshark.org/lists/wireshark-users/201005/msg00013.html

Biografi Penulis

Arsyad Dwiyankuntoko. Sedang menjalankan program D4 Teknik Telekomunikasi di Politeknik Negeri Semarang angkatan 2010

Copyright © 2003-2007 IlmuKomputer.Com